Movie Recommender System: Final Report

1. Introduction

This report documents the development of a movie recommender system. The system aims to suggest movies to users based on their demographic information and past movie preferences. Two approaches were explored: Singular Value Decomposition (SVD) and Linear Regression (LR). This report covers data exploration, solution implementation, training process, and evaluation against a benchmark.

2. Data Analysis

The MovieLens 100K dataset, consisting of 100,000 ratings from 943 users on 1682 movies, was used. The dataset includes user demographics (age, gender, occupation, zip code) and movie genres.

Key Observations

- **Ratings Distribution**: Most ratings range between 3 and 4, indicating a positive trend (Figure 1).
- **User Demographics**: Predominantly younger users (20s-30s), with more males than females. Diverse occupations, with a high number of students and educators (Figure 2-4).
- **Movie Genres**: Drama and Comedy are the most prevalent genres (Figure 5).

Preprocessing Implications

- **Demographics**: Age categorization and one-hot encoding for gender and occupation.
- **Movie Genres**: One-hot encoding for genre categorization.
- **Handling Sparse Data**: Techniques like matrix factorization for the sparse user-item interaction matrix.

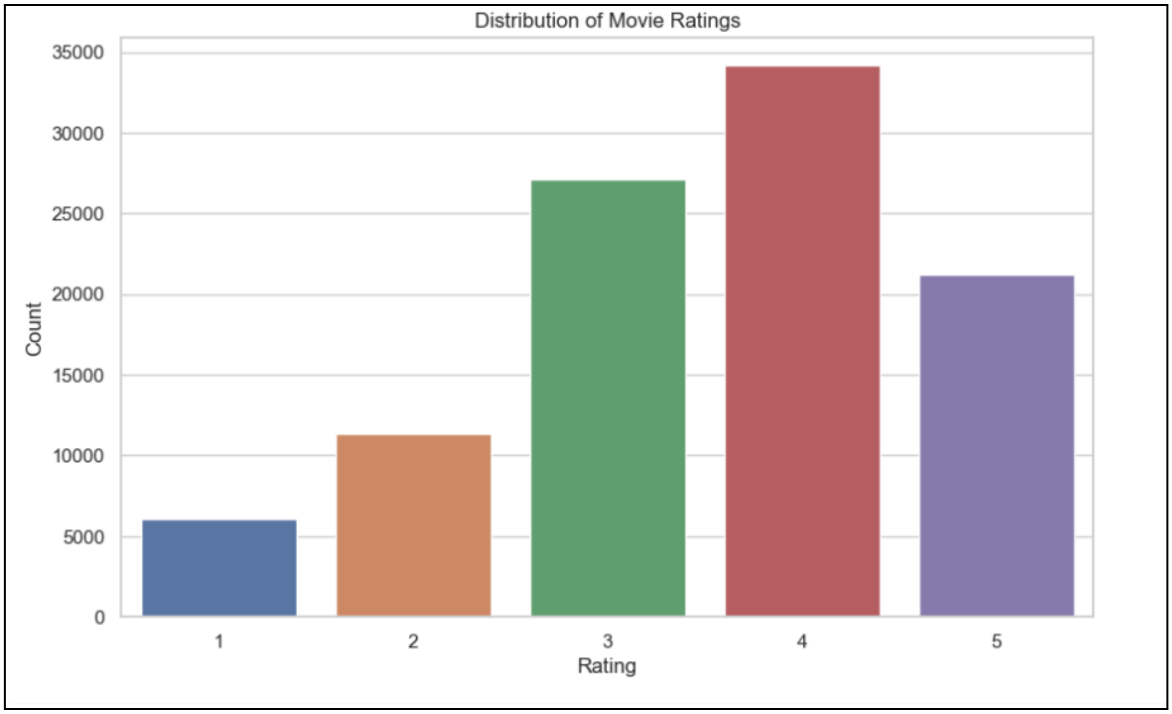

Figure 1. Distribution of Movie Ratings

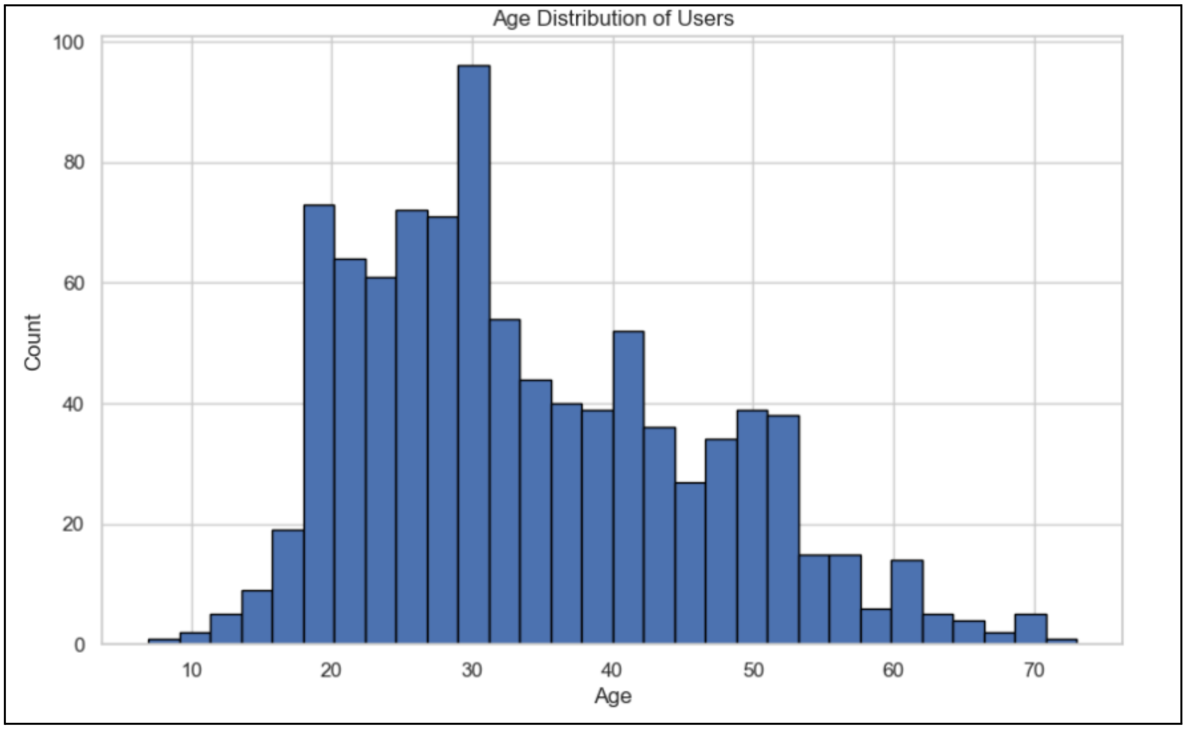

Figure 2. Age Distribution of Users

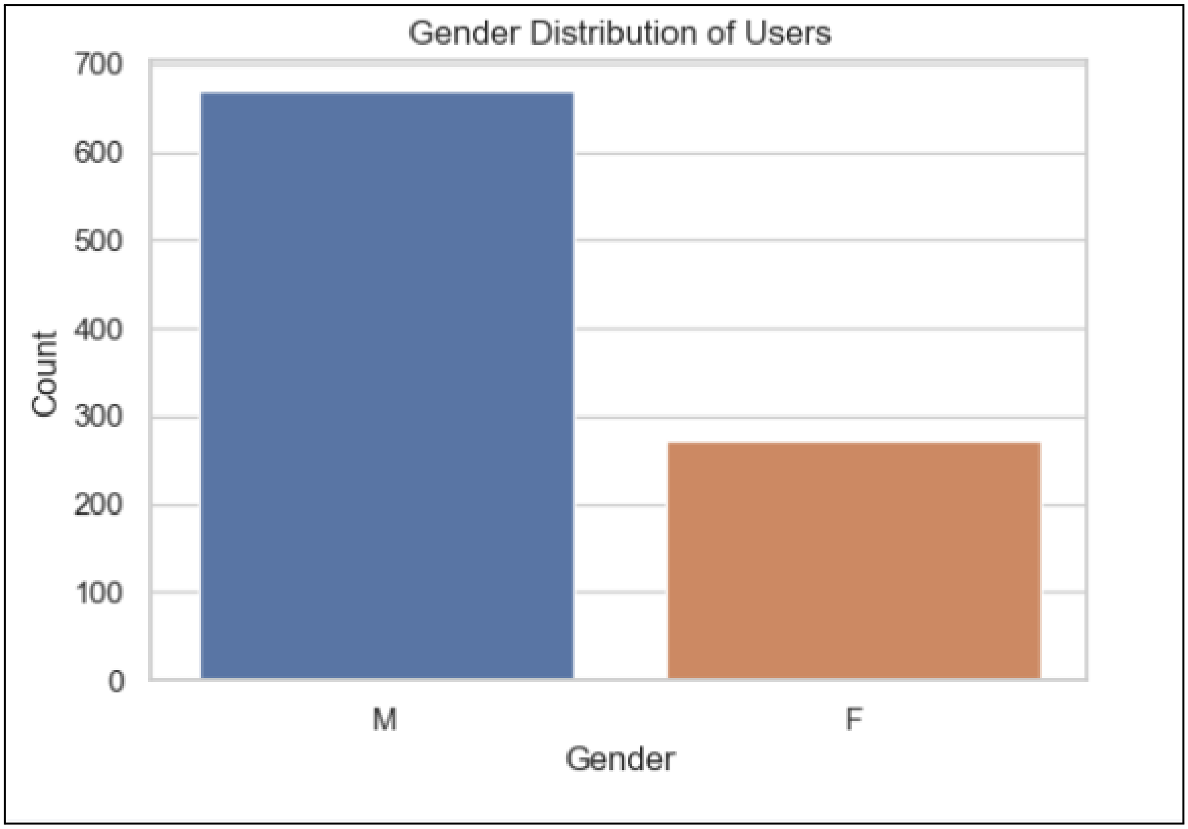

Figure 3. Gender Distribution of Users

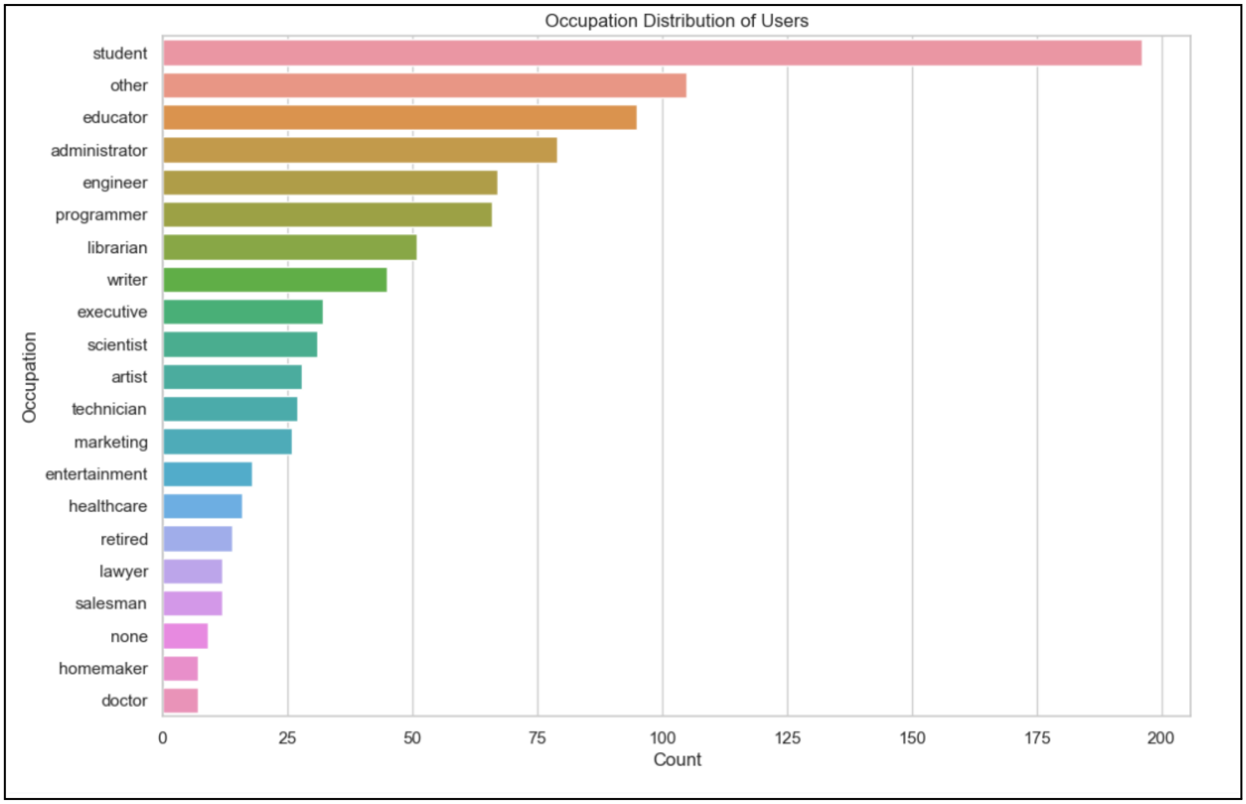

Figure 4. Occupation Distribution of Users

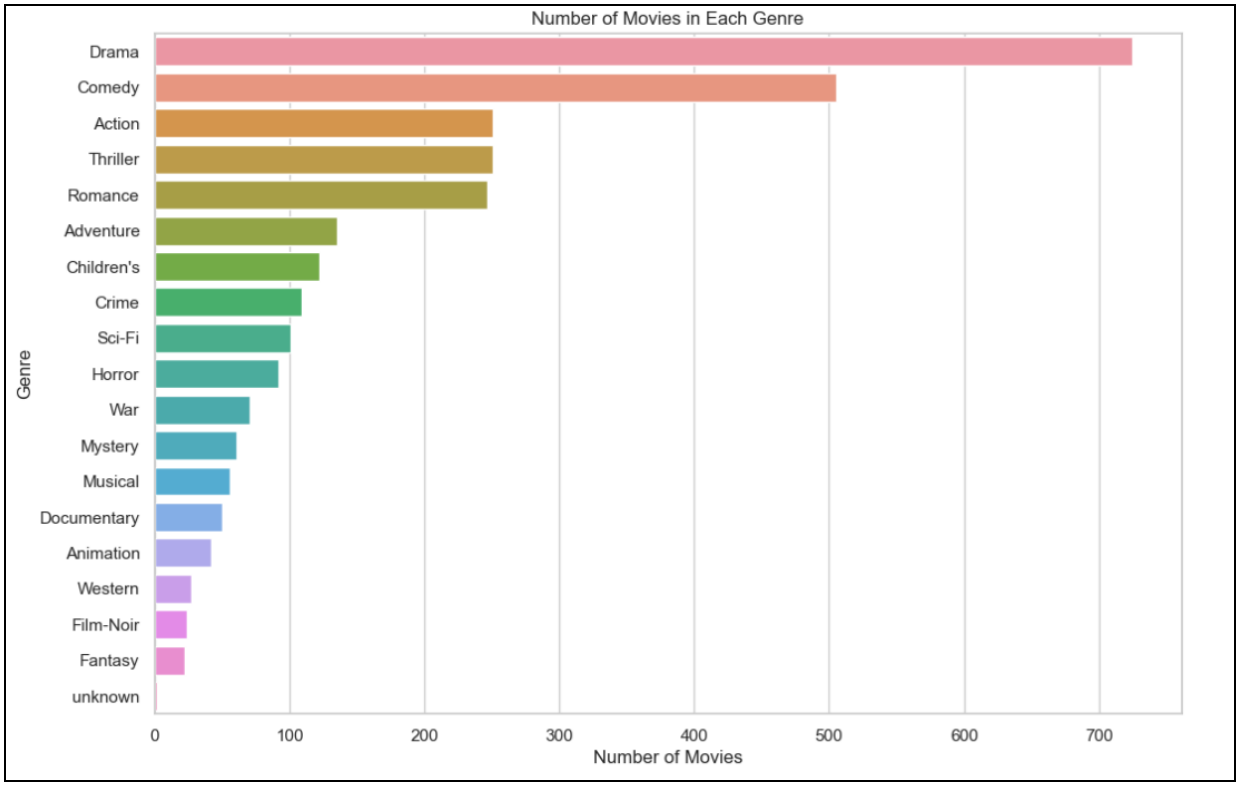

Figure 5. Number of Movies in Each Genre

3. Models Implementation

Two models were implemented:

1. **SVD-Based Recommender**:
   - Utilized SVD for matrix factorization.
   - Predicted user ratings for movies based on latent factors.
   - Capable of handling sparse data effectively.

2. **Linear Regression-Based Recommender**:
   - Used demographic and movie genre data to predict user ratings.
   - One-hot encoding for categorical variables.
   - Normalized age feature.
   - *ElasticNet* regression was chosen for its ability to combine L1 and L2 regularization.

4. Models Advantages and Disadvantages

- **SVD Advantages**: Good at capturing latent factors, effective with sparse data.
- **SVD Disadvantages**: Requires dense matrices, sensitive to overfitting with many latent factors.
- **Linear Regression Advantages**: Interpretable, can incorporate demographic data.
- **Linear Regression Disadvantages**: May not capture complex user-item interactions, assumes linear relationships.

5. Training Process

- **SVD Model**: Trained on user-item matrices, varying the number of latent factors for optimization. *The best obtained number of latent factors = 10.*
- **Linear Regression Model**: Trained on user demographics and movie genres. Regularization parameters were tuned for optimal performance. *The best obtained hyperparameters: alpha = 1; l1_ratio = 0.5.*

6. Evaluation

Evaluation Benchmark

The models were evaluated using “Hit Ratio @ 10”, a common metric for recommender systems [1]. This metric measures whether the test set's actual high-rated movie appears in the top 10 recommendations.

Protocol for Hit Ratio @ 10 Calculation:

1) **Item Selection**: For each user, 99 items that the user has not previously interacted with are selected randomly. These items are combined with a ”test item”, which is an item the user has actually interacted with, resulting in a set of 100 items. Besides, this “test item” should be highly rated by user.
2) **Model Prediction and Ranking**: The recommender model is then applied to these 100 items, and a prediction score is assigned to each item. The items are subsequently ranked based on their predicted scores.
3) **Top 10 Analysis**: From the ranked list of 100 items, the top 10 items are selected. If the 'test item' (the item with actual user interaction) is found within these top 10 items, it is classified as a 'hit.'

4) **Aggregated Measure Across Users**: This process is repeated for all users in the test dataset. The 'Hit Ratio @ 10' is computed as the average of these hits across all users.

Moreover, Hit Ratio @ 10 was computed on several train/test sets provided in the dataset (u1.base, u1.test, u2.base, u2.test, etc.). “Test item” (film that was seen and highly rated by the user) was always chosen from test sets to ensure this information was not used during training. Finally, the average Hit Ratio @ 10 was computed across all provided train/test sets.

Achieved Results

- **SVD Model Benchmark:**

  Average Hit Ratio @ 10: **0.82**.

- **Linear Regression Model Benchmark:**

  Average Hit Ratio @ 10: **0.68**.

7. Results

SVD Model significantly outperformed Linear Regression Model. Using SVD Model, we achieved very good results despite the fact this method is quite simple and computationally cheap.

8. Related works

[1] James Loy, “Deep Learning based Recommender System”: https://www.kaggle.com/code/jamesloy/deep-learning-based-recommender-systems/notebook